The TL@NTU Text-to-speech System for the Blizzard Challenge 2019

Wenjie Li[1], *Haihua Xu*[2],*Eng Siong Chng*[23]

[1]School of Information Science and Engineering, Xinjiang University, China
[2]Temasek Laboratories @ Nanyang Technological University Singapore
[3]School of Computer Engineering, Nanyang Technological University (NTU), Singapore

liwenjie@mipitalk.com, hhx502@gmail.com

Abstract

We describe the TL@NTU's text-to-speech system for Blizzard Challenge 2019 in this paper. The target language of this year challenge is Mandarin, and some of the utterances to be synthesized contain English words, which actually belongs to a mixed-language text-to-speech task. Based on the above situation, we employ unit selection based waveform concatenation method in this year challenge, since we think it is easier to handle mixed-language text-to-speech issue, compared with the conventional statistical parametric method, which requires multilingual expertise to build an appropriate front-end text analyzer. We make efforts to build the waveform concatenation system mainly focusing on two aspects. Firstly, we are building flexible phone unit search table, allowing for approximate context-phone vector search. This is crucial for the system built with insufficient data. Secondly, we propose a simplified waveform concatenation method, yielding improved synthesized results.

Index Terms: text-to-speech,unit selection, waveform concatenation,

1. Introduction

Text-to-speech(TTS) synthesis is a technique to generate speech waveform from a given input text [1]. In recent years, TTS synthesis technology has been developed rapidly. This begins with using Deep Neural Network (DNN) to substitute conventional GMM-HMM models to estimate acoustic features from the input linguistic features.

The process of DNN based Statistical Parametric Speech Synthesis (SPSS) [2] approach is straightforward. It can be realized with the following steps. The first step is to do text analysis to generate the front-end linguistic features. This requires a huge linguistic expertise. Meanwhile, the acoustic features, which are employed to synthesise the sound in the following steps, are to be extracted and aligned to the corresponding linguistic features. After that, we are to train a DNN model mapping the linguistic features to the corresponding acoustic features. Finally, a vocoder is employed to synthesise the speech taking the acoustic features from the DNN as input.

Recently, WaveNet [3], an auto-regressive waveform generative model, produces high quality synthesised speech that is comparable to the real human speech [4]. Such a technique has already been widely used in many TTS [5, 6, 7] and voice conversion tasks [8, 9, 10]. Technically, given different conditional input features, WaveNet behaves differently. For instance, if WaveNet is employed as a jointly learnable TTS synthesizer, the conditional features would be linguistic features [3], as the conventional input of the DNN method. In contrast, if WaveNet is employed as a learnable vocoder, the conditional features could be different acoustic features, such as mel spectrogram or filterbank features [11, 12]. Since WaveNet is an auto-regressive model, originally it is generating waveform sample by sample, the process of generation is extremely slow. To address the issue, various kinds of speed-up approaches such as Parallel Wavenet [13] and Clarinet [14] are proposed. Though a WaveNet is learnable, potentially producing higher quality voice, one of its drawbacks lies in its highly necessity of human-crafted linguistic features, as well as higher quality waveform to train the models.

More recently, an end-to-end text-to-speech approach named Tacotron [15] greatly simplifies the speech synthesis process. It employs attention mechanism [16] to do forced-alignment between spectrograms and letter/character sequences implicitly. Once the end-to-end models generate the spectrograms, with which one can uses the Griffin-Lim algorithm [17] to estimate the waveform/voice. Interestingly, with the spectrograms as conditional features, one can also use the WaveNet as aforementioned to synthesize the voice.

Despite significant progress on text-to-speech technique development in recent years, one still has challenges to synthesize high quality speech, particularly in the case of insufficient or low quality training data. This is what one faces in this year Blizzard Challenge. The organizer only releases about 8 hours of training data in Mandarin, and the data format is MP3 which is lossy format. Besides, the data itself contains Mandarin-English mixed utterances. If we want to synthesize speech for such utterances, a bilingual front-end parser should be off the shelf. This is also a challenge for us. Based on such considerations, we choose the conventional unit selection based waveform concatenation method addressing the challenge.

Different from the previous TTS methods as mentioned, unit selection based waveform concatenation TTS method [18] belongs to non-parametric one. To build a waveform concatenation TTS system, one has to solve the following problems. 1) To begin with, a waveform unit must be determined, and a database of such units must be built [19]. 2) Design an algorithmic strategy to search the predefined unit[20, 21]. 3) Choose an algorithm to realize waveform concatenation [22, 23, 24, 25, 26]. 4) Post-processing can be optionally employed to smooth the concatenated waveform reducing the glitch. In this paper, we are focusing on the first three problems, of which problems 2) and 3) are emphasized specifically.

In this paper, we use phone units, which are the initials and finals of the syllable for Mandarin and normal phone for English, to build search database. To alleviate context-phone sparsity issue due to the incompleteness of the context-phone coverage of the training data, we propose a search strategy, considering both exact and approximate match. Since the approximate match based search method can potentially yield a lot of loosely relevant units, concatenating such units would produce degraded synthesized speech. As a result, we propose a simple but effective unit concatenation method, which results in

acceptable results with such limited training data.

2. TL-NTU unit selection synthesis system

2.1. Phone unit table building and search

The most priority task for unit selection based waveform concatenation is to define key-value tables realizing phone unit search as accurate and flexible as possible. In this work, the keys are integer vector representing context-phone information. The value is simple, including wave file ID, both start time and end time of each phone. For clarity, we also call such a value as a candidate waveform unit. Each values are unique the in the training data. The key problem is how to choose those candidate waveform units correctly. As a result, the most important task is how to design a key vector representing a phone unit. Our guideline is to be exact and flexible. That is, when a context phone appears in the training data, we should search it precisely, when a context phone is absent, we should find a substitute phone with a gentle relaxation, yielding minimal voice degradation. Based on the above consideration, our key vector is defined as in Table 1.

Table 1: *Phone unit search key definition, where each key represents as a integer vector. With different value for each component, it can realize both exact and inexact phone unit search,"Comp." refers to the component of the key vector.*

	Comp.	Meaning
Key	c_1	Previous phone ID
	c_2	Present phone ID
	c_3	Next phone ID
	c_4	Present phone forward position in word
	c_5	Present phone backward position in word
	c_6	Word ID
	c_7	Word forward position in utterance
	c_8	Word backward position in utterance

To realize exact search using the phone unit key as shown in Table 1, we just assign each component with the real integer value corresponding to the position of each word and phone. This is straightforward.

However one has many options realizing inexact search following the key definition in Table 1. The simplest one is to ignore some components when conducting search. For instance, one can ignore component c_6 representing "word ID", or other phone ID, etc. In our work, we choose another alternative, where the the components of the integer vector are defined differently, as is shown in Table 2.

As shown in Table 2, the "position" of each phones and words is not precisely determined in our work, and it has three alternatives, namely, the start, intermediate, and the end of each word or utterance. In practise, we found the key as defined in Table 2 yields better results for inexact match based search.

We note that after results submission, we realized both Table 1 and Table 2 have flaws. They are not considering the word ID of the neighboring phones. This can yield inexact match at the word boundary when we are meant to only focus the exact match.

Table 2: *Phone unit search key definition, only for inexact match, "Comp." refers to the component of the key vector.*

	Comp.	Meaning
Key	c_1	Previous phone ID
	c_2	Present phone ID
	c_3	Next phone ID
	c_4	Present phone position in present word
	c_5	Present word ID
	c_6	Present word position in utterance

2.2. Waveform concatenation

If each candidate waveform unit selected from the tables as mentioned in Section 2.1 is unique, then waveform concatenation is a trivial task. However, the difficulty lies in for each candidate phone we have many candidate waveform units. Therefore, waveform concatenation is crucial for a quality TTS system.

Right now, as the training data is insufficient, context-phone coverage is significantly incomplete. Besides, our phone unit selection algorithm itself is not perfect, yielding unnecessary noisy candidate units. As a result, the objective of the waveform concatenation method is to select those appropriate units that are minimizing the concatenation cost. To achieve this objective, we are meant to resolve two problems. First, we use tandem acoustic features which are more discriminative. Secondly, we propose a simple but effective method to estimate concatenation cost with the tandem features.

The tandem features include 14-dimensional KALDI [27] MFCC plus pitch features [28], 63-dimensional WORLD acoustic features [29] including MGC (60-dim), log F0 (1dim), AP (1-dim), and 30-dimensional bottleneck features [30] respectively. The bottleneck feature extractor is trained with KALDI MFCC features.

With the extracted features, we propose a method to estimate the concatenation cost as illustrated in Figure 1.

The basic assumption of our proposed method is as follows. Suppose each candidate of the selected waveform set for each context-dependent phone are equally correct itself, then if the two consecutive candidates are able to concatenate, theoretically, the concatenation cost for the two candidates should be zero. Specifically, we estimate the cost as follows. Suppose the window length that is used to estimate the cost is N (N=2, as illustrated in Figure 1). We first extract N features starting from the end of the first phone, then we extract N features staring from the beginning of the second phone (Theoretically, if they are real consecutive, then the two sequences are actually overlapped). We compute averaged Euclidean distance between two feature sequences as cost C_1, which we call it as forward-extended cost in Figure 1. Similarly, we compute the cost C_2, which we name it as backward extended cost in Figure 1. We can think of them as "symmetric". As a result, the final cost is $C_1 + C_2$.

2.3. Data preparation

Aside from the focus on the algorithmic part as mentioned in Section 2.1 and Section 2.2, data preparation is also an important procedure for a TTS system development. First, the transcription of this year challenge are very noisy, except for English words, it contains a lot of Arabic numbers and special symbols, and different Chinese punctuation. We conduct

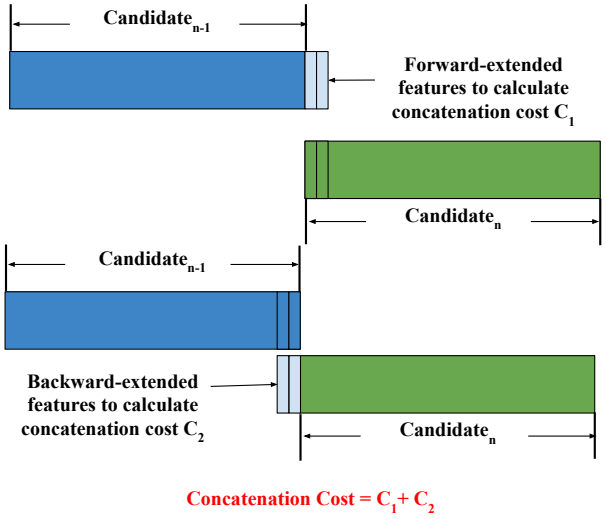

Figure 1: *Using forward-backward extended feature sequences to estimate concatenation cost*

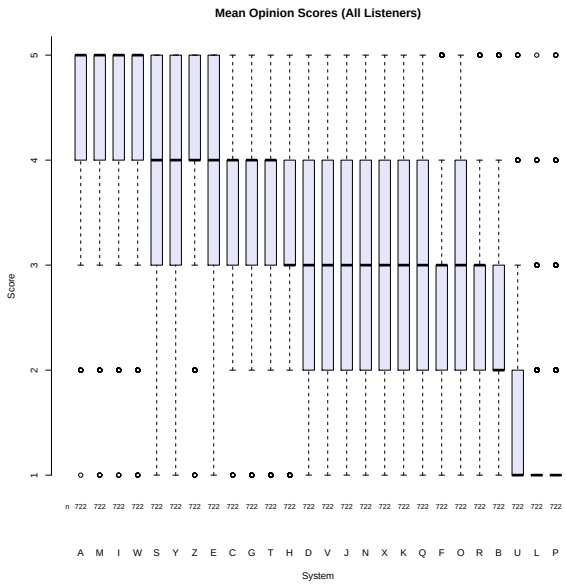

Figure 2: *Mean Opinion Scores (MOS) results from all listeners, our system is represented with "Q".*

a series of text normalization work, till all tokens in the transcription are either English or Mandarin words. For audio file preparation, we first convert the MP3 files into 16kHz wave files. We also conduct speech volume normalization, ensuring all the speech volume of the wave files stable. After these, we use our in-house English-Mandarin code-switching ASR system [31] to do forced-alignment to get the time-boundary for each phone and corresponding word. In practice we extract the context-phone key and corresponding waveform unit as value from the alignment at the same time. Finally, we employ various kinds of methods to extract different acoustic features and concatenate them to estimate concatenation cost.

3. Evaluation Results

This section presents the evaluation results released by the organizer. Our system is labelled as "Q in this year challenge.

Three kinds of listeners participated in the listening tests, including paid listeners, online volunteers and speech experts.

There are 26 systems in evaluation test including 1 benchmark baselines, 24 participant teams and a natural speech. System A is a natural speech, the System B is the DNN benchmark built using the the Merlin toolkit [32].

Four types of evaluation were conducted in this year, including Mean opinion scores(MOS), Similarity with original speaker(SIM), Pinyin (without tone) Error Rate (PER) and Pinyin (with tone) Error Rate (PTER). the MOS evaluates the naturalness of the synthetic sentence with a score scale of 1 to 5. The SIM represents how similar the synthetic voice is close to the reference samples on a scale from 1 to 5. PER and PTER represent the accuracy of the synthetic speech, the difference is PER doesn't consider tone's error,and both of PER and PTER evaluated with word error rate (WER).

Figure 2 shows the overall Mean Opinion Score (MOS) results, where our system is represented as "Q".

From Figure 2, the overall performance of our system is worse than majority of competitive systems. Our system only performs better than five systems,include the Merlin baseline system as indicated with "B". The limitations of our system mainly come from two aspects. First, its approximated search results in phone units with different tone. Additionally, its cost of concatenation is too simplistic, yielding a lot glitches. The most obvious advantage is its capability to handle the mix language TTS, while baseline Merlin system cannot handle that without bilingual front-end facility.

Figure 3 reports the overall similarity scores presented by the entire listeners between the synthesized speech and the original speech.

The results shown in Figure 3 make us slightly surprised. Although our system is a unit selection based waveform concatenation method, the similarity score of our system is still left behind some systems such as "M" or "S". We guess this is again due to the two aforementioned reasons: 1) the synthesized waveform by our system contains a lot of glitches, which can produce degraded results. 2) our inexact match based phone unit search method could ignore tone, which leads to undesired tonal variation in the synthesized speech, as a result it yields worse similarity.

Figures 5 and 4 present the accuracy of the synthesized system in Toned and Untoned syllable error rates respectively.

Our system is even worse than baseline,As discussed,we think the glitches and inexact match also had a negative impact on the WER results. Here, the glitched could have more severe effect.

4. Discussion & Conclusion

So far, there are two main options to build a TTS system. One is statistical parameter based method, such as DNN, state-of-the-art WaveNet, end-to-end methods. The other is non-parametric method, such as unit selection based waveform concatenation method, as employed in this paper. However, without a down-to-earth sharper front-end text parser to generate linguistic features, it will be very challenging for one to build a desirable TTS system with limited and lower qua/lity training audio.

Originally, we intended to submit a statistical parameter

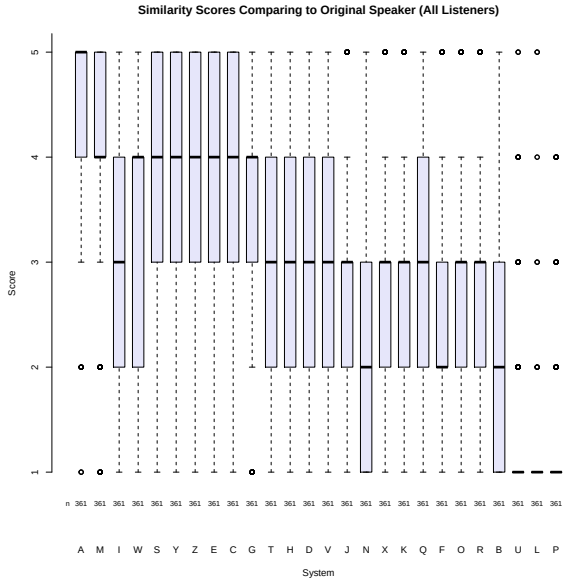

Figure 3: *Similarity scores from all listeners, comparing to the original speaker. Our system is represented with "Q".*

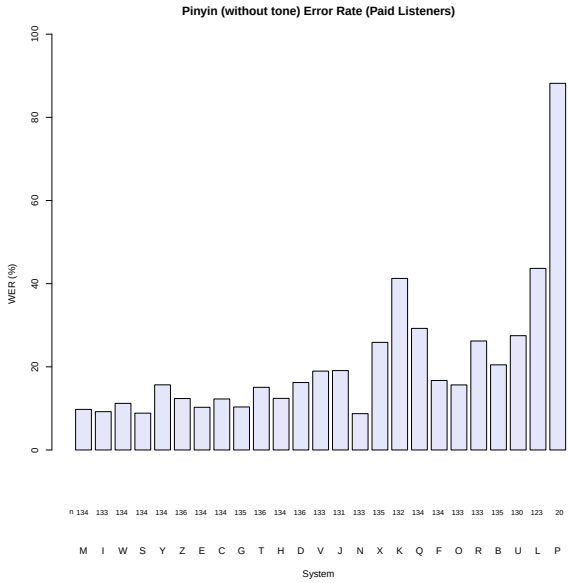

Figure 4: *Untoned Pinyin Error Rate for accuracy evaluation, our system is represented with "Q".*

based TTS system. However, we are short of a strong front-end text parser to generate good linguistic features. We found neither DNN, or state-of-the-art WaveNet can yield satisfactory results. Besides, we don't have many efforts by our own for building such systems, we finally gave it up.

We also tried end-to-end TTS method. Unfortunately, due to limited training data and the lower quality of the training data, we also failed to synthesize acceptable voices.

Eventually, we resort the unit selection based waveform concatenation method built over KALDI platform. Due to our rough phone unit based search method, waveform concatenation method, as well as lack of a post-smoothing method, the overall system performance is severely affected. In future, we are ameliorating them.

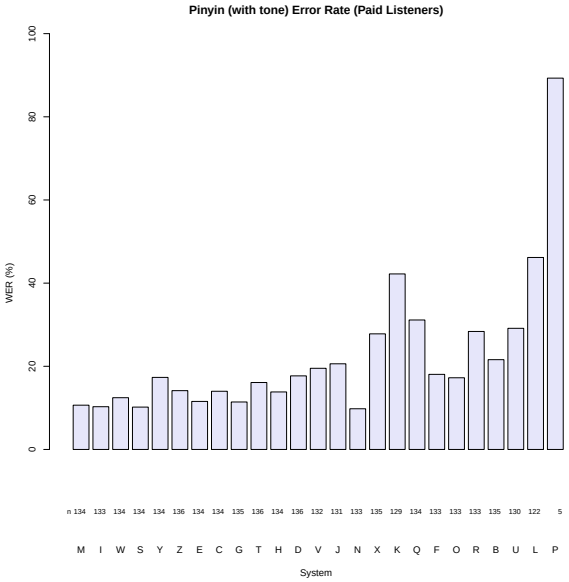

Figure 5: *Toned Pinyin Error Rate for accuracy evaluation, our system is represented with "Q".*

5. References

[1] S. King, "Measuring a decade of progress in text-to-speech," *Loquens*, vol. 1, no. 1, p. 006, 2014.

[2] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7962–7966.

[3] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[5] Z. Jin, A. Finkelstein, G. J. Mysore *et al.*, "FFTNet: A real-time speaker-dependent neural vocoder." 2018, pp. 2251–2255.

[6] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep voice: Real-time neural text-to-speech," pp. 195–204, 2017.

[7] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[8] K. Kobayashi, T. Hayashi, A. Tamamori, and T. Toda, "Statistical voice conversion with wavenet-based waveform generation." in *Interspeech*, 2017, pp. 1138–1142.

[9] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "Wavenet vocoder with limited training data for voice conversion." pp. 1983–1987, 2018.

[10] K. Chen, B. Chen, J. Lai, and K. Yu, "High-quality voice conversion using spectrogram-based wavenet vocoder." pp. 1993–1997, 2018.

[11] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent wavenet vocoder." pp. 1118–1122, 2017.

[12] N. Adiga, V. Tsiaras, and Y. Stylianou, "On the use of wavenet as a statistical vocoder," pp. 5674–5678, 2018.

[13] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[14] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[15] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[16] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," pp. 577–585, 2015.

[17] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[18] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.

[19] Z.-H. Ling and R.-H. Wang, "Hmm-based unit selection using frame sized speech segments," in *Ninth international conference on spoken language processing*, 2006.

[20] D. P. Bertsekas, "Dynamic programming and optimal control." *Belmont, MA: Athena scientific*, 1995.

[21] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[22] J. Vepa, "Join cost for unit selection speech synthesis." 2004.

[23] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis." in *in Proc. Eurospeech*, 1997.

[24] P. Taylor, "The target cost formulation in unit selection speech synthesis." in *Interspeech*, 2006, p. 20382041.

[25] Y. Qian, Z.-J. Yan, Y. Wu, F. K. Soong, X. Zhuang, and S. Kong, "An hmm trajectory tiling (htt) approach to high quality tts," 2010.

[26] M. Chu, P. Liu, Y. Zhao, and Y. Li, "Speech unit selection using hmm acoustic models," Mar. 6 2008, uS Patent App. 11/508,093.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[28] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2494–2498.

[29] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[30] F. Grézl and M. Karafiát, "Hierarchical neural net architectures for feature extraction in asr," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[31] H. Xu, V. T. Pham, Z. T. Kyaw, Z. hao Lim, E. S. Chng, and H. Li, "Mandarin-english code-switching speech recognition," in *Proc. of INTERSPEECH 2019*, 2018.

[32] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," in *Proc. of INTERSPEECH 2016*, 2018.